In the Claims:

Please amend claims 1, 4, 6, 7 and 9 as follows:

1. (Amended) A system for enabling code execution from non executable memory, comprising:

(i) an executing entity, for executing code for a host system;

(ii) a non-executable memory component, for storing system code and data; and

(iii) an executable memory component, for operating as a memory buffer for executing said code, such that a portion of contents of said non-executable memory component is located within said executable memory component, and said portion of contents of said non-executable memory component emulates executable functions of said executable memory component.

4. (Amended) A system for executing code using non-executable memory, comprising:

(i) an executing entity, for executing code;

(ii) a non-executable memory component, for storing said code and data; and

(iii) a plurality of executable memory components that alternate as memory buffers, thereby preventing memory lockage for accesses to said data during download operations of said code.

6. (Amended) A method for executing code using non-executable memory, comprising the steps of:

(i) providing executable memory, for buffering at least one code request from an executing entity;

(ii) providing a non-executable memory, for storing executable code;

(iii) downloading at least a portion of said executable code to said executable memory, for emulating executable functions of said executable memory;

(iv) executing at least one said code request from said executable memory; and

(v) buffering an execution of contents of said non-executable memory in said executable memory.

7. (Amended) The method of claim 6, further comprising the steps of:

(vi) managing at least one set of instructions to guarantee availability of said contents in an executable buffer; and

(vii) supplying a busy signal in cases where said contents are not available, such that the executing entity delays the read cycle until said contents are available.

9. (Amended) The method of claim 6, such that step (iv) further includes:

(a) providing a plurality of executable memory buffers for preventing said portion of said non-executable memory from being locked for accesses during said downloading operation;

(b) loading said executable code to one of said plurality of executable memory buffers; and

(c) maintaining at least one of additional said executable memory buffers, to be accessible to said executing entity and executable by said executing entity.

Please add new claims 10-22 as follows:

10. (New) A device for enabling an executing entity of a host system to execute code, comprising:

(i) a non-executable memory component, for storing the code; and

(ii) at least one executable memory component, each said executable memory component for presenting at least a portion of said stored code to the executing entity in a manner that enables the executing entity to execute said portion of said stored code directly from said each executable memory.

11. (New) The device of claim 10, wherein said non-executable memory component and said at least one executable memory component are separate from the host system.

12. (New) The device of claim 10, further comprising:

(iii) a mechanism for guaranteeing availability, in one of said at least one executable memory component, of code requested by the executing entity.

13. (New) The device of claim 10, comprising a plurality of said executable memory components, such that while one said executable memory component is presenting a first said at least portion of said stored code to the executing entity, a second said at least portion of said stored code is being downloaded to another said executable memory component.

14. (New) The device of claim 10, wherein each said at least one executable memory component is too small to accommodate all of the code at once.

15. (New) A method of executing code, comprising the steps of:

(a) storing the code in a non-executable memory component;

(b) downloading at least a portion of the code from said non-executable memory component to a first executable memory component; and

(c) executing said downloaded code, by an executing entity of a host system, said first executable memory component being separate from said host system.

16. (New) The method of claim 15, wherein said executing entity executes said downloaded code directly from said first executable memory component.

17. (New) The method of claim 15, wherein only a first portion of the code is downloaded to said first executable memory component.

18. (New) The method of claim 17, further comprising the steps of:

(d) subsequent to said downloading, requesting code to be executed, by said executing entity;

(e) if said requested code is outside of said downloaded first portion of the code:

(i) downloading a second portion of the code, including said requested code, from said non-executable memory component to said first executable memory component; and

(ii) during said downloading of said second portion of the code, suspending activity of said executing entity.

19. (New) The method of claim 18, wherein said suspending includes supplying a busy signal to said executing entity.

20. (New) The method of claim 16, further comprising the steps of:

(d) downloading a second portion of the code to a second executable memory component; and

(e) executing said downloaded second portion of the code, by said executing entity.

21. (New) The method of claim 20, wherein said second executable memory component is separate from said host system.